

State Duration and Interval Modeling in Hidden Semi-Markov Model for Sequential Data Analysis

Hiromi Narimatsu*

HiroYuki Kasai†

August 26, 2016

Abstract

Sequential data modeling and analysis have become indispensable tools for analyzing sequential data such as time-series data because a larger amount of sensed event data have become available. These methods capture the sequential structure of data of interest, such as input-output relationship and correlation among datasets. However, since most studies in this area are specialized or limited for their respective applications, rigorous requirement analysis on such a model has not been examined in a general point of view. Hence, we particularly examine the structure of sequential data, and extract the necessity of “state duration” and “state duration” of events for efficient and rich representation of sequential data. Specifically addressing the hidden semi-Markov model (HSMM) that represents such state duration inside a model, we attempt to newly add representational capability of state interval of events onto HSMM. To this end, we propose two extended models; one is interval state hidden semi-Markov model (IS-HSMM) to express the length of state interval with a special state node designated as “interval state node”. The other is interval length probability hidden semi-Markov model (ILP-HSMM) which represents the length of state interval with a new probabilistic parameter “interval length probability.” From exhaustive simulations, we show superior performances of the proposed models in comparison with HSMM. To the best of our knowledge, our proposed models are the first extensions of HMM to support state interval representation as well as state duration representation.

1 Introduction

The significant progress of portable devices and wearable devices with multi-functional sensors has enabled us to easily record all the sensing data and to record all the observed events and phenomena. This situation motivates us to analyze such recorded data, and many studies have explored a wide range of methods of pattern recognition, biological data analysis, speech recognition, image classification, behavior recognition, and time-series data analysis. Esmaili *et al.* categorized three types of sequential patterns after theoretical investigation for a large amount of data [1]. Lewis *et al.* proposed a sequential algorithm using queries to train text classifiers [2]. Song *et al.* proposed a sequential clustering algorithm for gene data [3]. More recently, studies using sensor data analysis for human behavior recognition and video sequence understanding have received significant attention because of the significant progress on wearable devices and the widespread of video surveillance systems [4, 5, 6]. Those devices enable users to record all of their experiences such as what is viewed, what is heard, and what is noticed. Nevertheless, although collecting all observed data

*H. Narimatsu is with the Graduate School of Information Systems, The University of Electro-Communications, 1-5-1 Chofugaoka, Chofu-shi, Tokyo, 182-8585, Japan (e-mail: narimatsu@appnet.is.uec.ac.jp)

†H. Kasai is with the Graduate School of Informatics and Engineering, The University of Electro-Communications, 1-5-1 Chofugaoka, Chofu-shi, Tokyo, 182-8585, Japan (e-mail: kasai@is.uec.ac.jp)

has become much easier, it remains difficult to immediately find the data that we want to access because the amount of time series data is extremely huge. In case of *life log* data application, for example, it must be much easy to exactly retrieve information of particular places or dates if rich and comprehensive *meta-data* are sufficiently attached to every piece of datum to be identified. However, if a query is very ambiguous like retrieving *a situation similar to the current situation where 10-minute continuous “Event A” starts 30 minutes later after half-hourly “Event B” finishes*, it must be surely challenging to obtain meaningful results at the end. Thus, finding such *similar sequential patterns* from vast sequential data using a given target pattern extracted from the current situation is of crucial importance. This is of interest in the present paper. Finding similar sequential patterns needs to discriminate particular sequential patterns from many *partial groups of multiple events* of patterns. For this purpose, among the specialized methods to detect similar partial patterns from sequential data, which include, for instance, Dynamic Programming (DP) matching algorithm, and Support Vector Machine (SVM), this paper dedicates solely to hidden Markov model (HMM) because HMM is specialized and promising to deal with sequential data by exploiting transition probability between states, i.e., events.

The primary contributions of our work are two-fold: (a) we advocate that the support of both “state duration” and “state interval” is of great significance to represent practical sequential data based on the studies about the feature and structure of sequential data, then we extract requirements for its modeling. Next, (b) we propose two new sequential models by extending hidden semi-Markov model (HSMM) [7] to support both state duration and interval of events efficiently. More concretely, regarding (a), we especially address the generalization of model requirements for sequential data, and emphasize the importance of handling events order, continuous time length, i.e., state duration of an event, and discontinuous time length, i.e., state interval between two events. To the best of our knowledge, this is the first generalization of the model requirements for sequential data. Herein, we define the continuous duration time of a state as state duration, and define the discontinuous interval with no observation as state interval, respectively because an event is treated as a state in HMM. Then, with respect to (b), after assessment of the extended HMM models in the literature against those requirements, we show that all the existing models do not treat both state duration and state interval simultaneously. Nevertheless, we also show that HSMM, one of the extended HMM models, handles state duration, and is an appropriate baseline to be extended to meet all the demands. Subsequently, in the present paper, we propose two extended models by extending HSMM that accommodates not only state duration but also state interval.

Two approaches are specifically addressed to treat state interval with HSMM. For both approaches, three variations can be considered to model state interval; modeling state interval with (i) only preceding state, (ii) only subsequent state, and (iii) both preceding and subsequent states. From the viewpoint of modeling accuracy, we specifically examine the modeling of state interval with both preceding and subsequent states. Then, we finally propose two extended models of HSMM: one represents state interval as a new *node of state interval*, and the other represents state interval by a new *probability of state interval length*. The first model, dubbed interval state hidden semi-Markov model (IS-HSMM), is categorized into a straightforward extension of the original HSMM. The distinct difference is the introduction of a new “interval state node.” A simple introduction of the interval state node into HSMM, however, leads to improper transition probabilities because the transition frequencies from ordinal state to the new interval state and the transition from the interval state to another state may increase when the symbols of interval state are observed frequently. This causes undesired biases onto the original transition probability, and finally brings severe degradation of model accuracy. To solve this issue, IS-HSMM expresses two-dimensional Markov model at the part where the preceding state is the interval state node. Meanwhile, the second proposed model is called as interval length probability hidden semi-Markov model (ILP-HSMM) which represents

state interval by a new parameter to HSMM. This parameter is “interval length probability” which is represented as a probability density distribution function, and is modeled with the two combined states. Preliminary studies of ILP-HSMM was proposed in our earlier work as DI-HMM [8].

The remainder of this paper is organized as follows. The next section introduces related work. Section 3 presents a description of the model requirements and the requirement assessment of the existing HMM variants. Then, a brief explanation of the original HMM model is given. Section 4 explains the baseline model of our proposal: hidden semi-Markov model, i.e., HSMM. After examining the approaches for state interval modeling based on HSMM in Section 5, we propose two models, IS-HSMM and ILP-HSMM, in Section 6 and Section 7, respectively. Finally, we show superior performances of the proposed models in comparison with HSMM in Section 8. We summarize the results presented in this paper and describe future work in Section 9.

2 Related Work

This section presents related work for sequential data analysis. For sequential pattern matching and sequential pattern detection, the Dynamic Programming (DP) algorithm [9] provides an optimized search algorithm that calculates the cost of a path in a grid and which thereby finds the least costly path. This was first used for acoustic speech recognition. For sequential pattern classification, Support Vector Machine (SVM) [10, 11] is a classifier that converts the n -class problem into multiple two-class problems. SVM has proved its superior performances in a wide range of applications such as face and object recognitions from a picture. As for Regression Model (RM)[12], the logistic regression model [13] is a representative model that is powerful binary classification model when the model parameters are independent each other. Hidden Markov model (HMM), originally proposed in [14, 15], is a statistical tool used for modeling generative sequences. HMM has been frequently used together with the Viterbi algorithm to estimate the likelihood of generating observation sequences. Whereas HMM is widely used for many applications, for example, speech recognition, handwriting recognition and activity recognition, many extended HMMs have been also proposed to enhance the expressive capabilities of the baseline HMM model as well as to support various specialized application data. Thus, addressing HMM is a powerful and robust model for treating sequential data using its transition probability in a statistical manner, we particularly examine HMM in the present paper.

With regard to the extensions of HMM, Xue *et al.* proposed transition-emitting HMMs (TE-HMMs) and state-emitting HMMs (SE-HMMs) to treat the discontinuous symbol [16], of which application is an off-line handwriting word recognition. The observation data contain discontinuous and continuous symbols between characters when writing in cursive letters. They specifically examined such discontinuous features and continuous features, and extended HMM to treat both of them. Bengio *et al.* focused on mapping input sequences to the output sequences [17]. The proposed model supports a recurrent networks processing style and describes an extended architecture under the supervised learning paradigm. Salzenstein *et al.* dealt with a statistical model based on Fuzzy Markov random chains for image segmentations in the context of stationary and non-stationary data [18]. They specifically examined the observation in a non-stationary context, and proposed a model and a method to estimate model parameters. Yu *et al.* proposed the explicit-duration hidden Markov model [19, 20, 21, 22]. They emphasized the interval between state transition, and proposed a new forward-backward algorithm to estimate the model parameters.

Addressing the difference of duration in each state, hidden semi-Markov Model (HSMM) is proposed to treat the duration and multiple observations produced in one single state [7, 23]. HSMM has been applied for some application data like time-series data. Bulla proposed an estimation

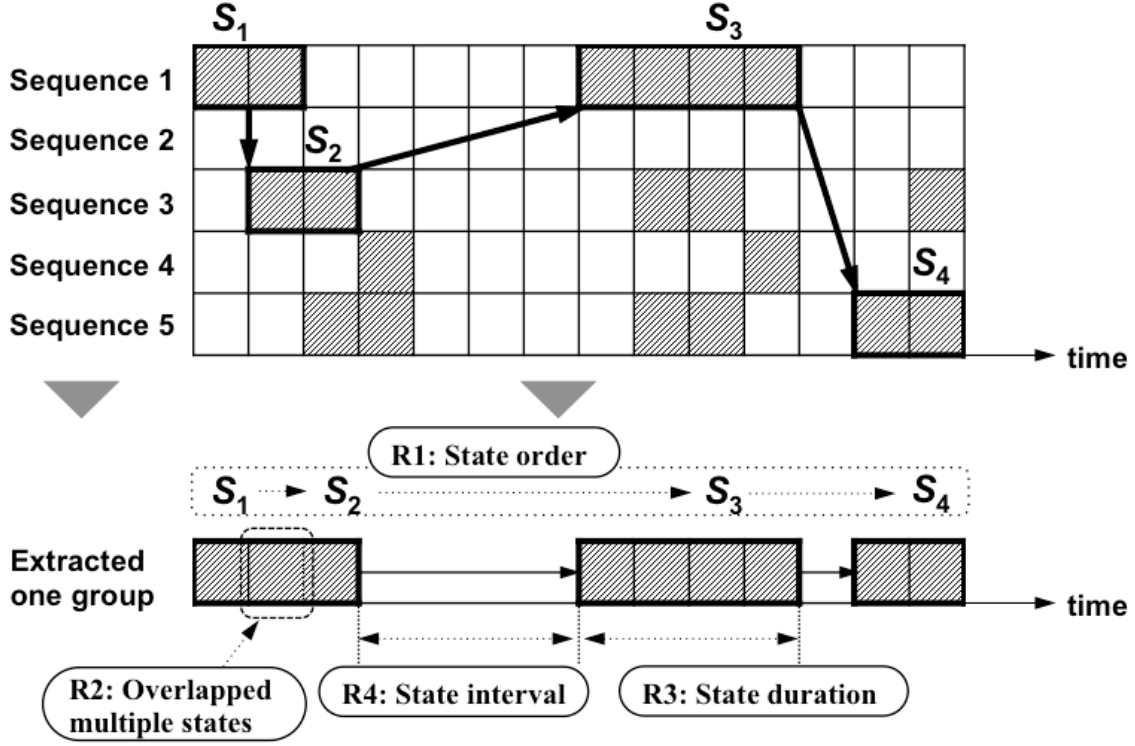


Figure 1: Event generative model and sequential data model requirements.

procedure to the right-censored HSMM for modeling financial time-series data using conditional Gaussian distributions for the HSMM parameters [24]. For diagnosis and prognosis using multi-sensor equipment, Dong *et al.* prioritized the weights for each sensor to treat multiple sensor results, and showed that the proposed model of HSMM gave higher performances than the original HSMM [25]. Recently, Dasu analyzed the technique of HSMM, and described how to implement HSMM for a practical application in detail [26].

3 Analysis of Sequential Data Modeling

This section presents an analysis of sequential data modeling and derives the model requirements for sequential data analysis. Then, the satisfactions of the extended models of HMM for the model requirements are examined.

3.1 Requirement for Model Description

This section presents discussions on the requirements for model description using time-series data: representative data of sequential data. For this purpose, we assume the situation where multiple different sequences are independently generated from five sensors as shown in Figure 1. Here, an observed event of which value exceeds a predefined threshold is recognized as a ‘state’ represented in a block. The continuous period of each event is represented by the length of the block. Since events are not successively observed, namely, *no-observation period*, exists between two successive states in certain periods, the length of such no observation period is represented as the distance

Table 1: Requirement satisfactions in HMM, HMM variants, and our proposals.

Model	Requirements	
	Time length in a state (R3)	Time Interval between states (R4)
HMM (baseline) [27]		
HMM-selftrans [16]	✓	
HSMM [7, 23]	✓	
FO-HMM [18]		
EDM [19, 20]	✓	
IO-HMM [17]		
IS-HSMM and ILP-HSMM (proposal)	✓	✓

between two blocks. In this example, we also assume that a set of four black-colored blocks, $\{S_1, S_2, S_3, S_4\}$, express an extracted multiple states that forms one particular group.

Now we extract the requirements for model description. First, addressing this formation of four blocks, we see that these states are surely observed in a prescribed order. Thus, we find that the order of multiple states should be described in a model (**R1**). Second, multiple states can be observed in a partially overlapped manner as seen in S_1 and S_2 . In other words, multiple states can occur simultaneously at a certain period. Therefore, the model needs to support the representation capability to describe multiple states occurring at the same time (**R2**). Third, since the time lengths of respective states mutually differ, the duration of a state needs to be expressed in a model (**R3**). Finally, for the case in which each state occurs intermittently, a vacant period between one state and another state that is not involved in the group of sequence might exist between two states. Furthermore, the length of this vacant period shall be variable. Therefore, state interval between two states in a model must be described (**R4**). In summary, the sequential data model is required to describe these requirements, and this paper defines these respective requirements as follows;

- (i) **R1**: State order
- (ii) **R2**: Staying multiple states in a certain period
- (iii) **R3**: State duration
- (iv) **R4**: State interval

Among these items, **R2** differs from other items because **R1**, **R3**, and **R4** are required even for a single sequence while **R2** is the requirement for multiple sequences. Therefore, this study specifically examines requirements **R1**, **R3**, and **R4**. The examination of **R2** shall be left for advanced researches to be undertaken in future work.

3.2 Requirement Verification for Extended HMM Models

This section investigates whether HMM and the extended variants of HMM satisfy those requirements. Table 1 summarizes a comparison among the existing HMM models from the viewpoints of the model requirements described above. Since the baseline HMM model describes the order of the states (**R1**), all the extended HMM models inherit this capability. FO-HMM is specialized for treating the ambiguity of observation symbols, and does not contribute to our model requirement. IO-HMM is a hybrid model of generative and discriminative models to treat the estimation probability commonly used for input sequence and observations. Therefore, it does not satisfy the

remaining requirements. HSMM models the time length to remain in one single state [7], and its variants including HMM-selftrans and EDM [19, 20, 21, 22] satisfy the same requirements: state order (**R1**) and state duration (**R3**).

As a result of investigation of the requirement satisfaction, we find that no existing HMM model accommodates both state duration and state interval together. Nevertheless, we conclude that HSMM is the best baseline model to be extended towards our new target model because only HSMM handles state duration. Hence, the next section gives a detailed explanation of HSMM.

4 Hidden Semi-Markov Model (HSMM)

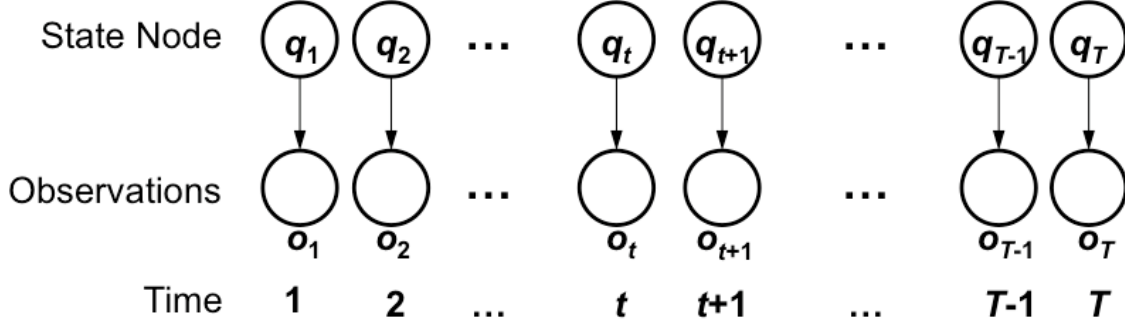
HMM has been studied as a powerful model for speech recognition because the important parts for modeling are the features of each phoneme and the order of phonemics [28, 29, 21]. The model parameters of HMM consists of the initial state probability, the transition probability between states and the emission probability of observation elements from each state. Whereas the model training phase calculates the optimum values of the model parameters, the recognition phase calculates the probabilities for each model that generates a observed sequence when the sequence is observed from each model using the trained model parameters, and finally selects the most appropriate model as a recognition result.

Among the model parameters, the distinguishing feature of HMM is to model the transition probability of every pair of two states. However, the time length to stay in each state is also essential for modeling in some useful applications like online handwriting recognition. For this purpose, HSMM has been proposed to support this time length, and has long been studied for some specific applications such as speech recognition and online handwriting recognition. This section, after providing basic notations, details the algorithms of the model training and recognition in HSMM.

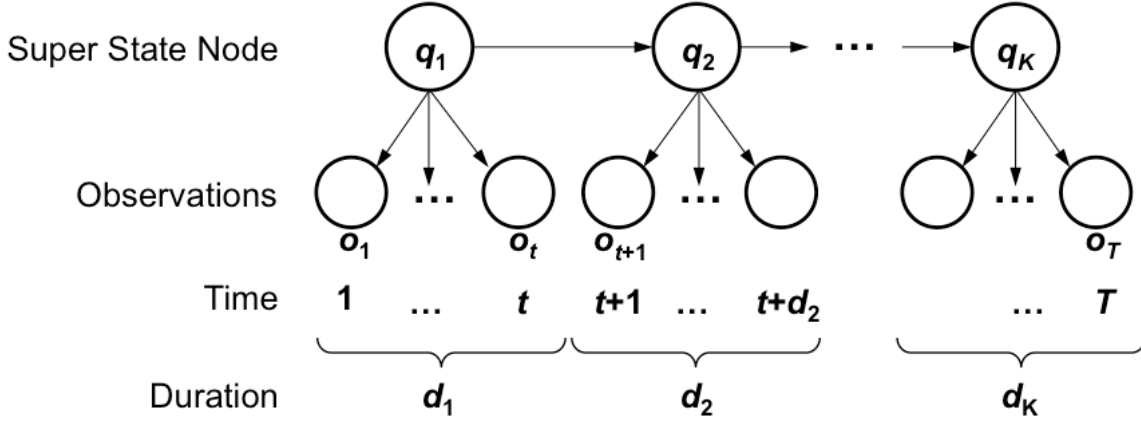
4.1 Notations

The HSMM structure is shown in Figure 2 comparing with that of HMM. Hereafter, we assume that each unit time at time t has one corresponding observation o_t . The observation sequence from time $t = t_1$ to $t = t_2$ is denoted as $\mathbf{o}_{t_1:t_2} = o_{t_1}, \dots, o_{t_2}$. A set of output symbols that are actually observed is expressed as $Y = \{y_1, y_2, \dots, y_N\}$, where N is the number of observations. Therefore, o_t can be expressed as $o_t = y_n$, where $y_n \in Y$. A set of hidden states is $S = \{1, \dots, M\}$, where M is the number of states' kinds, and the hidden state sequence from time $t = 1$ to $t = T$ is expressed $S_{1:T} = S_1, \dots, S_T$, where S_t represents a state at time t . While HMM allows each state node to emit an observation symbol, HSMM allows each super state node to emit multiple observation symbols, i.e, observation sequence. Here the hidden state sequence is represented as $\mathbf{Q} = q_1, \dots, q_k, \dots, q_K$ where K is the number of state in a sequence. While $K = T$ in HMM, $K \leq T$ in HSMM. Then, the k -th hidden state is assigned a state i as $q_k = i (i \in S)$ in both HMM and HSMM as well. The parameters incorporated in the HMM model are $\Lambda = \{\mathbf{A}, \mathbf{B}, \boldsymbol{\pi}\}$, where $\mathbf{A} \in \mathbb{R}^{M \times M}$ is the matrix representing the transition probabilities between states, $\mathbf{B} \in \mathbb{R}^{M \times N}$ is the matrix for the emission probability from each state, and $\boldsymbol{\pi} \in \mathbb{R}^M$ represents the initial probability values that occur each state. In general, the transition probability from state i to state j is denoted as $\mathbf{A}(i, j) = a_{ij}$ where $i, j \in S$. Similarly, the emission probability of symbol y_n from state j is represented as $b_j(y_n)$ and $\mathbf{B}(j, n) = b_j(y_n) = b_j(o_t)$ where $j \in S$ and $y_n \in Y$. The initial probability of occurring each state is denoted as $\pi_i \in \boldsymbol{\pi}$.

On the other hand, HSMM handles the same set of parameters $\Lambda = \{\mathbf{A}, \mathbf{B}, \boldsymbol{\pi}\}$, but the elements of each parameter differ from those of HMM to describe the duration of states. The set of duration



(a) Model structure of HMM. The state node of HMM emits an observation symbol.



(b) Model structure of HSMM. The super state node of HSMM emits observation sequence while a certain duration.

Figure 2: Model structure comparison between HMM and HSMM.

time is denoted as D and the duration of state i is represented as $d_i \in D$. Considering this new parameter, the transition probability from state i to state j is represented as $a_{(i,d_i)(j,d_j)}$ in stead of $a_{i,j}$, and the emission probability is represented as $b_{j,d_j}(o_{t+1:t+d_j})$ in stead of $b_j(o_t)$. The parameters Λ is updated by the recursive calculation for inference and the latest calculation result for update is represented as $\hat{\Lambda}$. The overall algorithm is summarized in Algorithm 1.

4.2 Model Training (Inference)

This section presents a description of how to train the model of HSMM using training sequences, i.e., how to estimate the set of parameters $\Lambda = \{\mathbf{A}, \mathbf{B}, \boldsymbol{\pi}\}$ including the duration in each element. In general, HSMM trains the model using Baum-Welch algorithm [14] as the same way as HMM, where a recursive forward-backward algorithm is used. The forward-backward algorithm is an inference algorithm used for HMM, and the an extended algorithm special for HSMM is also proposed[30].

The concrete calculation algorithm for HSMM is detailed as follows: computing forward probabilities starts from $t = 1$ to $t = T$, and computing backward probabilities from $t = T$ to $t = 1$. This two-way calculations repeat until the amount of increase from the previous likelihood to the updated likelihood converges below a predefined threshold. More concretely, the forward step calculates the

Algorithm 1 Algorithm for training and recognition in HSMM.

Require: Input

Training sequences : $\mathbf{o}_{1:T_r}^z = \{o_1^z, \dots, o_{T_r}^z\}$,

Testing sequences : $\mathbf{o}_{1:T_t}^* = \{o_1^*, \dots, o_{T_t}^*\}$.

Ensure: Training phase

```

1: for  $z = 1$  to  $Z$  do
2:   Assign random values to the HSMM parameters  $\Lambda^z = \{A, B, \pi\}$ , and  $\alpha_{t(j,d_j)}$  and  $\beta_{t(j,d_j)}$ .
3:   for  $h = 1$  to  $H$  do
4:     for  $t = 1$  to  $T_r$  do
5:       Calculate  $\alpha_{t(j,d_j)}$  and  $\beta_{t(j,d_j)}$  using (1) and (2).
6:       Update parameters  $\Lambda^z$  using (3) and (4).
7:     end for
8:     Calculate  $\theta_h$  using (5).
9:     if  $\theta_h - \theta_{h-1} < \epsilon$  then
10:      break
11:    end if
12:  end for
13: end for

```

Ensure: Recognition phase

```

14: for  $z = 1$  to  $Z$  do
15:   for  $t = 1$  to  $T_t$  do
16:     Prepare  $\Lambda^z$  from the results obtained in the training phase.
17:     Calculate  $\alpha_t(j, d_j)$  using (6).
18:   end for
19:   Calculate  $P(o_{1:T_t} | \Lambda^z)$  using  $\alpha_t(j, d_j)$ .
20: end for
21: Select the model  $z^*$  that has the maximum value for  $P(o_{1:T_t} | \Lambda^z)$ .
22: Return Model  $z^*$  and its probability  $P(o_{1:T_t}^* | \Lambda^{z^*})$ .

```

following forward variable $\alpha_t(j, d_j)$ of state j with d_j at t as

$$\alpha_t(j, d_j) = \sum_{i \in \{S\} \setminus \{j\}} \sum_{d_i \in D} \alpha_{t-1}(i, d_i) a_{(i,d_i)(j,d_j)} b_{j,d_j}(\mathbf{o}_{t-d_j+1:t}). \quad (1)$$

The backward step calculates the following backward variable $\beta_t(j, d_j)$ as

$$\beta_t(j, d_j) = \sum_{i \in \{S\} \setminus \{j\}} \sum_{d_i \in D} a_{(j,d_j)(i,d_i)} b_{i,d_i}(\mathbf{o}_{t+1:t+d_i}) \beta_{t+d_i}(i, d_i). \quad (2)$$

The calculation step for estimating model parameters are summarized below.

Step 1 Initialization

Give an initial set of parameters Λ of the model at random.

Step 2 Recursive calculation

Calculate the set of parameters $\hat{\Lambda}$ that maximizes the variables of the forward-backward algorithm using the initialized parameter Λ . Denoting the updated state transition probability a and the updated emission probability b as a' and b' , respectively, $a'_{(i,d_i)(j,d_j)}$ and $b'_{j,d_j}(\mathbf{o}_{t+1:t+d_j})$ are updated using the previous values of $a_{(i,d_i)(j,d_j)}$ and $b_{j,d_j}(\mathbf{o}_{t+1:t+d_j})$. More specifically, the state transition

probability from state i with d_i to state j with d_j is defined as

$$a_{(i,d_i)(j,d_j)} := P(S_{t+1:t+d_j} = j | S_{t-d_i+1:t} = i).$$

Analogous to the state transition probability, the emission probability of $\mathbf{o}_{t+1:t+d_j}$ from state j with d_j is defined as

$$b_{j,d_j}(\mathbf{o}_{t+1:t+d_j}) := P(\mathbf{o}_{t+1:t+d_j} | S_{t+1:t+d_j} = j).$$

Then, these probability updates are calculated as (3) and (4) using the variables of (1) and (2) as

$$a'_{(i,d_i)(j,d_j)} = \frac{\sum_{t=1}^T \alpha_t(i, d_i) a_{(i,d_i)(j,d_j)} b_{i,d_i}(\mathbf{o}_{t+1:t+d_i}) \beta_{t+1}(j, d_j)}{\sum_{t=1}^T \alpha_t(i, d_i) \beta_t(i, d_i)}. \quad (3)$$

$$b'_{j,d_j}(\mathbf{o}_{t+1:t+d_j}) = \frac{\sum_{t=1}^T \delta(o_t, y_n) \alpha_t(j, d_j) \beta_t(j, d_j)}{\sum_{t=1}^T \alpha_t(j, d_j) \beta_t(j, d_j)}, \quad (4)$$

where $\delta(o_t, y_n)$ is defined as

$$\delta(o_t, y_n) = \begin{cases} 1 & \text{if } o_t = y_n \\ 0 & \text{otherwise.} \end{cases}$$

Step 3 Parameter update and log-likelihood calculation

Update the set of parameters as $\Lambda = \hat{\Lambda}$ using the result of **Step 2**. Calculate the probability that outputs the observation sequence $\mathbf{o}_{1:T}$ from the current model, and finally calculate the log-likelihood as

$$\hat{\theta} = \log P(\mathbf{o}_{1:T}) = \log \sum_{j=1}^M \alpha_T(j, d_j), \quad (5)$$

where $\hat{\theta}$ is the updated log-likelihood probability.

Step 4 Convergence judgement

Judge whether the estimation process converges by evaluating that the amount of increase from the previous likelihood θ to the updated likelihood $\hat{\theta}$ in **Step 3** is less than a predefined threshold ϵ as

$$\hat{\theta} - \theta < \epsilon.$$

If the condition above is satisfied, the process is terminated, otherwise **Step 2** and **Step 3** are iterated until the amount of increase converges.

4.3 Recognition using HSMM

For the recognition phase that finds the model that is most likely to generate a given target observation sequence, the probability of generating observation sequence plays an essential role. For this purpose, we first assume that a *label* is appropriately assigned into each group of sequence in advance, and the recognition step is defined to seek the most suitable label for a given group of sequence by calculating the label of the model that has the maximum probability as a recognition result. The probability of generating the target observation sequence is calculated using the forward algorithm used in HMM. For each model, it recursively calculates the forward variable and the probability for each state using $P(\mathbf{o}_{1:T}) = \sum_{i=1}^M \alpha_T(i, d_i)$, which is the marginal probability distribution, where

$$\alpha_t(j, d_j) = \left[\sum_{i=1}^M \alpha_{t-1}(i, d_i) a_{(i, d_i)(j, d_j)} \right] b_{i, d_i}(\mathbf{o}_{t-d_i+1:t}). \quad (6)$$

Here, we denote explicitly the probability as $P(\mathbf{o}_{1:T}^* | \Lambda^z)$ using the parameter set of model z , i.e., Λ^z , where $z \in \{1, 2, \dots, Z\}$ and Z is the total number of models. Finally, the label that has the maximum $P(\mathbf{o}_{1:T})$ for the observation sequence is selected as the recognition result. Thus, the model z^* that has the maximum probability $P(\mathbf{o}_{1:T}^* | \Lambda^{z^*})$ among all Z models is selected as the result of the recognition.

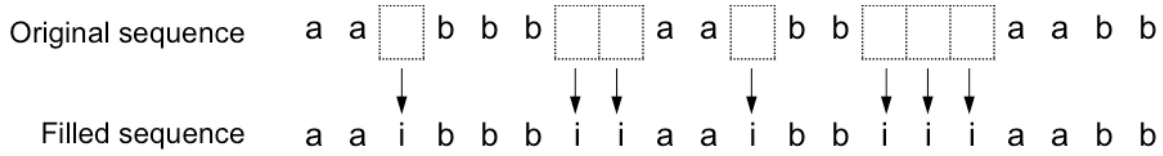


Figure 3: Representation of state interval in a sequence.

5 State Interval Modeling in HSMM

This section investigates how to model state interval in a model using HSMM. Before explaining the details, we describe how to represent state interval in a sequence. The baseline HSMM model ignores the period when no event is observed because the occurrence of events and the order of the events are essential for sequential data modeling. However, we also take into consideration this period i.e., no-observation period, because this is also essential to model sequential data as described in Section 3.1. Therefore, we regard this period as state interval in this paper, and assign a new symbol “interval symbol” to this period. Figure 3 illustrates an example of the state interval representation, where “a” and “b” are symbols actually observed in the original sequence, and “i” is the interval symbol used to fill state interval. Section 5.1 examines the approaches for modeling state interval using HSMM, and the issues that arise due to the filled sequence with state interval are addressed in Section 5.2.

5.1 Two Approaches for State Interval Modeling

To treat state interval with HSMM, two approaches can be considered as illustrated in Figure 4. One represents state interval as a new state node, which is represented as a black node as Figure 4

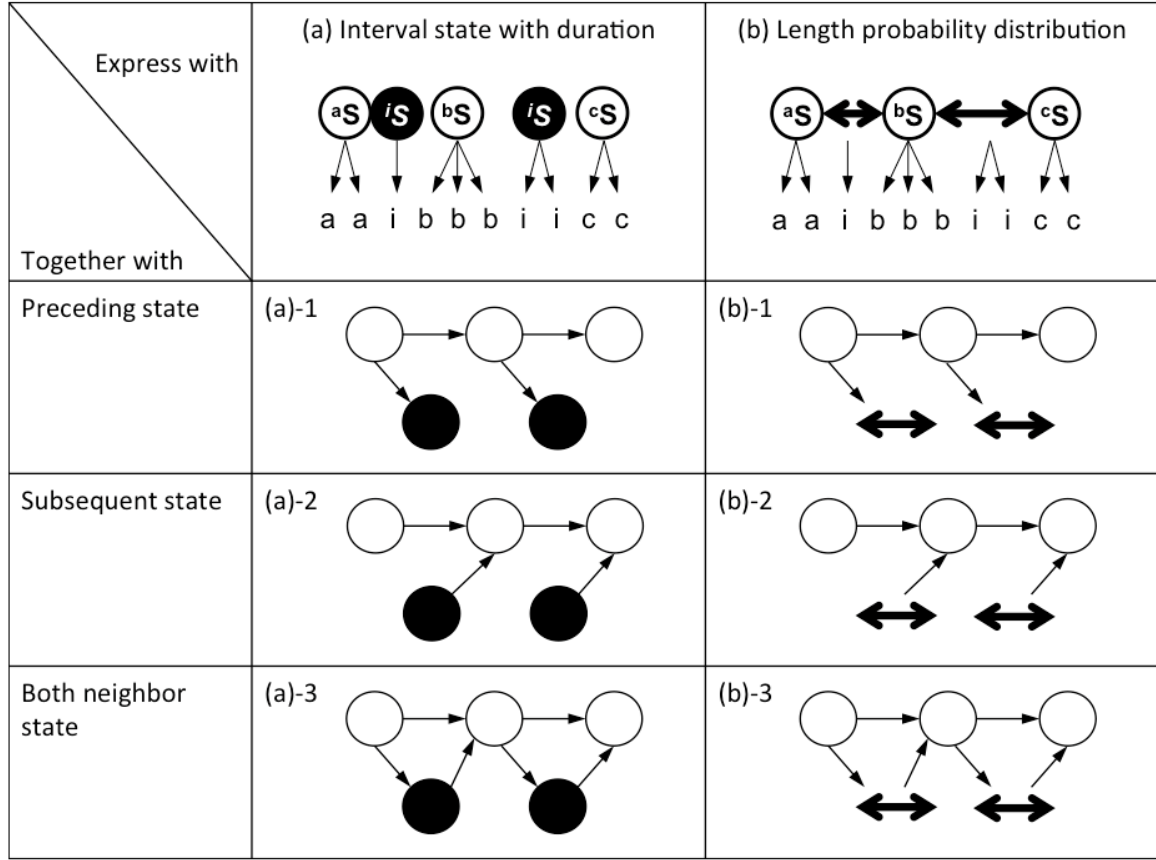


Figure 4: Two approaches for state interval. Circle represents a state and \rightarrow represents the transition from the left state to the right state. Circle filled with black and \leftrightarrow represent state interval.

(a). Because each state of HSMM can represent its duration staying in one single state, this new approach describes the length of state interval by introducing the new state node that explicitly indicates state interval. On the other hand, the other approach represents state interval as an new probabilistic parameter as shown in Figure 4 (b).

For both approaches, three variations to model state interval can be considered. The first approach models state interval with the preceding state ((a)-1, (b)-1), and the second models it with the subsequent state ((a)-2, (b)-2). The last variation models the length of interval with both preceding and subsequent states ((a)-3, (b)-3). Compared among three variations, the first two models have connection with only one state whereas the last one ((a)-3, (b)-3) has connections with two states. Therefore, (a)-3 and (b)-3 can model the sequential data more precisely.

5.2 Problems of State Interval Modeling

Before describing the proposed models, the technical issues for the state interval modeling in each approach in the preceding subsection are explained. The structure of the first approach is illustrated in Figure 5, where the interval state node is represented as a black node iS . Although this approach handles state interval in a simple way, it causes big bias in the transition probability when there are many groups of terms observed interval symbols in a sequence as shown in Figure 6. Details of the problem are examined using an example presented in the same figure. Figure 6 (a) shows

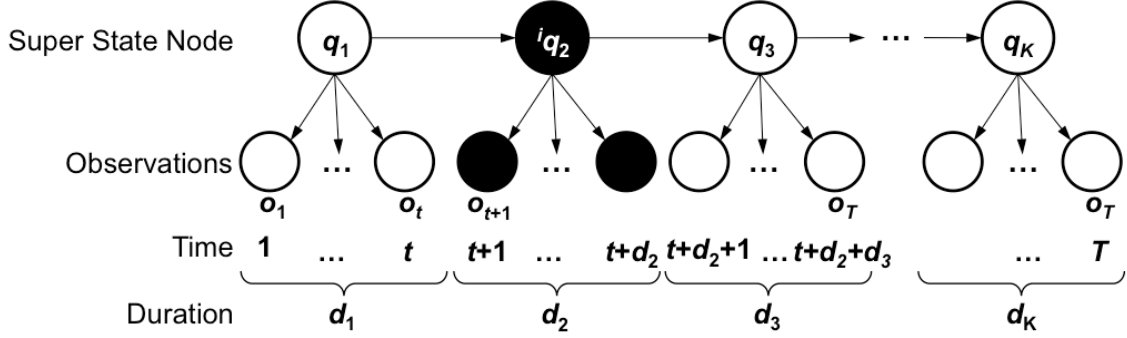


Figure 5: HSM with interval state.

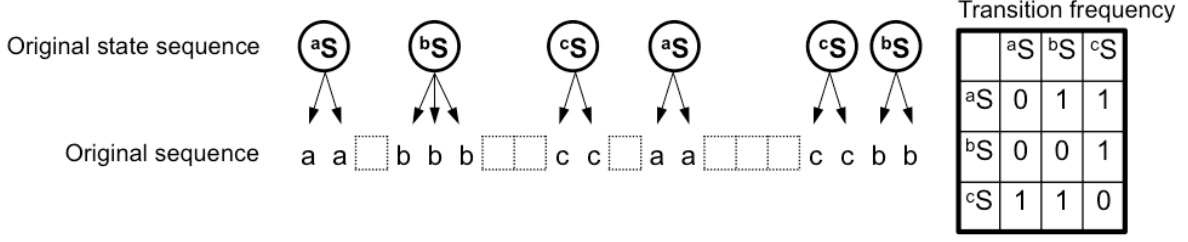
an example sequence for the explanation. Each sequence shows the original observation sequence and the state sequence. On the other hand, Figure 6 (b) shows an example sequence filled with state interval nodes of interval symbol i . The tables represented at the right of the figure show the transition frequency from a state to another state calculated using the original/complemented sequence. While the states described in a vertical line in the table show the “from” states, the states in a horizontal line show the “to” state. The table in (a) shows the transition frequency calculated using the original state sequence, and the table in (b) shows the transition frequency calculated using the converted state sequence filled with interval states. Accordingly, the results reveal that the transition frequency in the while cells except for gray painted cells falls dramatically to lower level, i.e., nearly zero. This means that, the introduction of the interval state node causes a deviation to the original transition probability, and the resultant new model fails to represent the transition sequence properly.

On the other hand, for the second approach in the preceding subsection, the manner of representing state interval with the new probabilistic parameter “interval length probability” must be defined. Considering application data, the model is expected to be found such that sequential data have a similar sequential pattern with similar state duration and state interval. Hence, it is required to model state duration and state interval with representing the similarity of its time length. Therefore, the second approach defines how to represent the new parameter for state duration and how to model the parameters with original HSMM in a probabilistic way.

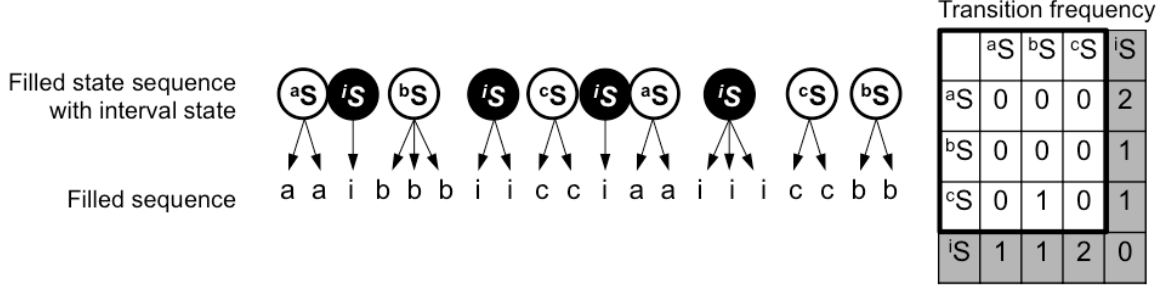
Addressing these problems, we finally propose two extended models in the following sections; interval state hidden semi-Markov model (IS-HSMM) as the first approach, and interval length probability hidden semi-Markov model (ILP-HSMM) for the second approach.

6 Interval State Hidden Semi-Markov Model (IS-HSMM)

HSMM handles state interval in a simple way because the interval symbol is replaced with the new interval state node as described in Section 5. However, we face the problem of the degradation of the accuracy of the transition probability in cases where state intervals appear frequently in the same sequence. To solve this problem, we propose an extended model, IS-HSMM, to preserve the transition probability of the original sequence. Figure 7 illustrates a conceptual structure of IS-HSMM. For easy-to-understand explanation, we pick up the first half three states shown in Figure 7 as an example when q_1 and q_2 are original hidden states and q_3 is the interval state node. While the original HSMM infers the transition probability in the order of q_1 , $i q_2$, and q_3 , the proposed IS-HSMM infers the transition probability to q_3 using two transition probabilities not only from $i q_2$



(a) Original sequence and transition frequency.



(b) Filled sequence and transition frequency.

Figure 6: Problem of sequence with state interval.

to q_3 but also from the previous q_1 to q_3 in order to preserve the transition of the original sequence. This is the noteworthy feature of IS-HSMM. This section provides how to train and how to recognize using the model as follows.

6.1 Model Training in IS-HSMM

The difference against the baseline HSMM model appears in the calculation of the forward variables and backward variables in the recursive calculation step. The state transition probability from state i to state j where the interval state iS is inserted between state i and state j is defined as

$$a_{(i,d_i)(iS,i d)(j,d_j)} := P(S_{t+1:t+d_j} = j | S_{t+1:t+i d} = iS, S_{t-d_i+1:t} = i),$$

where the duration of interval state iS is denoted as $i d$, and the durations of state i and j are d_i and d_j , respectively. The forward variable is calculated as

$$\alpha_t(j, d_j) = \sum_{i \in \{S\} \setminus \{j\}} \sum_{d_i \in D} \alpha_{t-1}(i, d_i) a_{(i,d_i)(iS,i d)(j,d_j)} b_{j,d_j}(\mathbf{o}_{t-d_j+1:t}). \quad (7)$$

Then, the backward variable is expressed as

$$\beta_t(j, d_j) = \sum_{i \in \{S\} \setminus \{j\}} \sum_{d_i \in D} a_{(j,d_j)(iS,i d)(i,d_i)} b_{i,d_i}(\mathbf{o}_{t+1:t+d_i}) \beta_{t+d_i}(i, d_i). \quad (8)$$

Finally, the transition probability and the emission probability are updated using (7) and (8) by assigning the forward and backward variables obtained by (3) and (4).

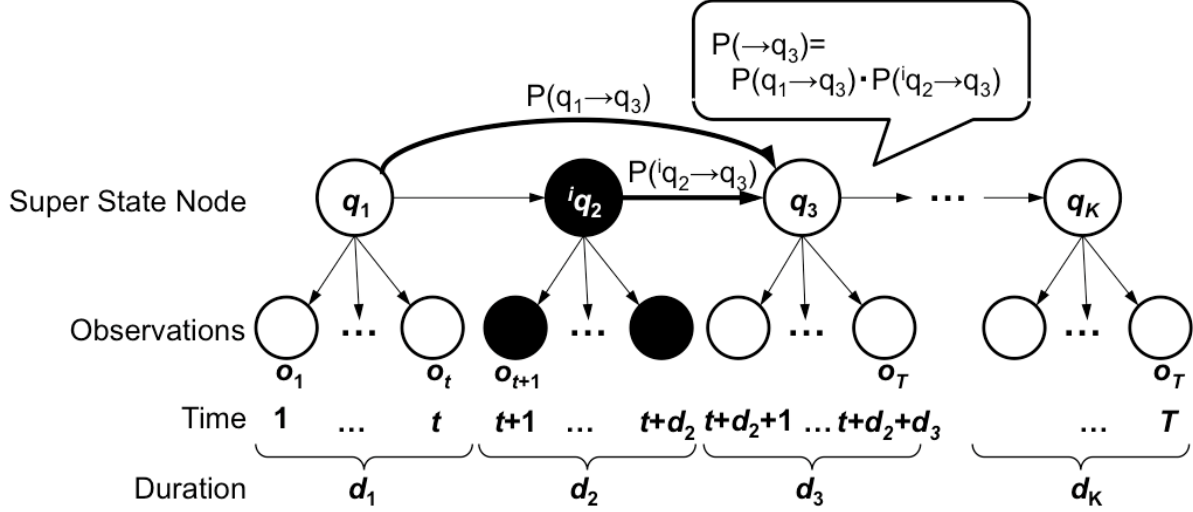


Figure 7: Conceptual structure of IS-HSMM with interval state node using two transition probabilities.

6.2 Recognition using IS-HSMM

While the calculation of the probability follows the original HSMM when the preceding state is not the interval state node, it differs when the preceding state is the interval state node. The probability of the observation sequence when the preceding state is the interval state node is calculated as $P(\mathbf{o}_{1:T}) = \sum_{i=1}^M \alpha_T(i, d_i)$, where

$$\alpha_t(j, d_j) = \left[\sum_{i=1}^c \alpha_{t-1}(i, d_i) a_{(i, d_i)(i_s, i_d)(j, d_j)} \right] b_{j, d_j}(\mathbf{o}_{t-d_j+1:t}). \quad (9)$$

The overall algorithm is presented in Algorithm 2.

7 Interval Length Probability HSMM (ILP-HSMM)

This section proposes ILP-HSMM, that newly introduces interval length probability to the transition probability in order to handle state interval between two states. It should be noted that the interval length probability corresponds to the probability density distribution of interval length of two states, to be technically precise. The distinct difference between HSMM and ILP-HSMM is that, whereas state j starts just after the end time of state i in the original HSMM, state j starts after a length of time, $L_{i,j}$, passes since the end time of state i in ILP-HSMM. The conceptual model structure of ILP-HSMM is illustrated in Figure 8. Although the structure of ILP-HSMM is similar to that of HSMM illustrated in Figure 2, the interval length probability is newly added to HSMM as shown in Figure 8, where $L_{i,j}$ represents the time difference between the end time of state i and the beginning time of state j . Here, note that the total time length of the observation sequence T varies because of its dependency on the length of state duration and interval, leading to $T = \sum_{k=1}^K (d_k + l_{k-1,k})$, where $l_{k-1,k}$ is the time difference between the end of q_{k-1} and the beginning of q_k . The subsequent section describes how to model and how to recognize given datasets using ILP-HSMM.

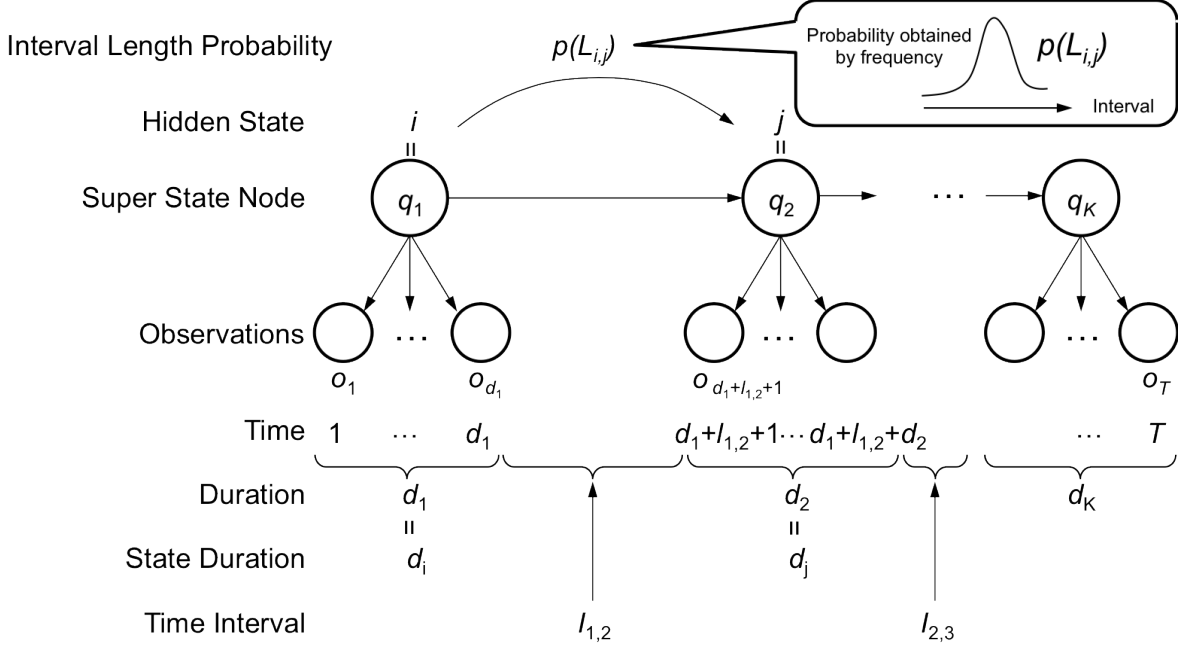


Figure 8: Conceptual structure of ILP-HSMM using state interval probability.

7.1 Model Training (Inference) in ILP-HSMM

Figure 9 shows an example data and representations used for explanation hereinafter. The slash line patterned blocks represent the data sequence of training dataset. First, the probability density distribution of the interval length of $L_{i,j}$ is expressed by the Gaussian distribution $p(L_{i,j})$ as

$$p(L_{i,j}) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(L_{i,j}-\mu)^2}{2\sigma^2}}, \quad (10)$$

where σ and μ represent the variance and the mean of $L_{i,j}$, respectively. It should be noted that the Gaussian distribution is adopted as the probability density distribution for simplicity. However, other distributions and functions could be adopted for ILP-HSMM without changing any other parameters. Accordingly, the set of parameters used in ILP-HSMM is defined as

$$\Lambda := \{\mathbf{A}, \mathbf{B}, \boldsymbol{\pi}, \mathbf{L}\},$$

where the elements of the parameter Λ take on $\mathbf{A}(i,j) = a_{(i,d_i)(j,d_j)}$, $\mathbf{B}(j,n) = b_{(j,d_j)}(\mathbf{o}_{1:d_j})$, and $\boldsymbol{\pi}(i) = \pi_{j,d_j}$, where $d_i \in D$ represents the duration of state i described in Section 4.1. Furthermore, $\mathbf{L} \in \mathbb{R}^{M \times M}$ is the matrix that consists of the interval length probabilities, i.e., the probability density distributions of state interval length, where $\mathbf{L}(i,j) = p(L_{i,j})$. The transition and emission probabilities are defined as the same as those in HSMM. The difference between HSMM and ILP-HSMM is to consider the parameter of $p(L_{i,j})$.

The range of $L_{i,j}$ in (10) might influence either memory consumption and/or computational complexity to generate the model. There might be no $L_{i,j}$ value suitable for the observation values because of the range limitation of $L_{i,j}$ if $p(L_{i,j})$ is generated in a training period. However, if the parameter $p(L_{i,j})$ is generated every time an observation is fed to the algorithm, the calculation cost can be much higher. Our motivation to introduce the interval length probability to HSMM is,

as explained earlier, to find the similar part of sequential data with respect to state interval and also to discriminate between the target part and the similar part. Therefore, even if the probability of $L_{i,j}$ is presumed to be zero around the skirts of the distribution, no critical problem arises. Consequently, we introduce the boundary of the probability value δ_{pt} to determine the edge of the skirt of $p(L_{i,j})$. On generating the $p(L_{i,j})$, the calculation is terminated when the probability value becomes less than δ_{pt} . The probability of $p(L_{i,j})$ is zero outside of the range of δ_{pt} .

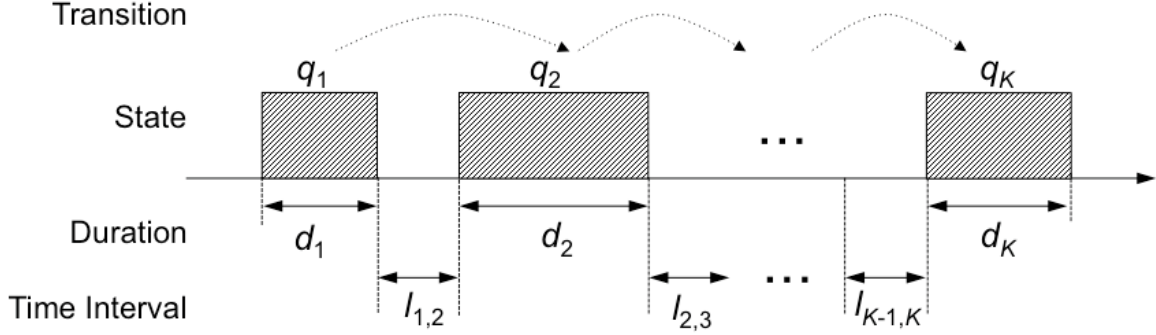


Figure 9: Sequential data and representations.

7.2 Recognition using ILP-HSMM

The Viterbi algorithm is used to estimate the probability of a model [22]. The pair of the model with the interval length probability and its label expected to be estimated are stored as the candidates for estimation. The recognition label that indicates the estimated result is selected when the model has the maximum likelihood estimate by calculating it for each state in each model.

First, we calculate $p(L_{i,j})$ beforehand. If $L_{i,j}$ is out of the range, the probability density distribution is determined as

$$p(L_{i,j}) = \min_{i,j \in S} p(L_{i,j}) \times c,$$

where c is $0 \leq c \leq 1$. Then, the forward variable for estimating the maximum likelihood is calculated as

$$\begin{aligned} \alpha_t(j, d_j) &:= \max_{s_{1:t-d_j}} P(s_{1:t-d_j}, s_{t-d_i+1:t} = j, \mathbf{o}_{1:d_j} | \Lambda) \\ &= \max_{i \in S \setminus \{j\}, d_i} \left\{ \alpha_{t-d_j}(i, d_i) \cdot a_{(i,d_i)(j,d_j)} \cdot b_{i,d_i}(\mathbf{o}_{t-d_i+1:t}) \cdot p(L_{i,j}) \right\}. \end{aligned} \quad (11)$$

The interval length probability is calculated simultaneously as calculating the parameter of the likelihood using the transition probability recursively.

The difference between HSMM and ILP-HSMM is the capability of handling the length of the state interval between states as explained earlier. The interval length probability in ILP-HSMM could be integrated by introducing each interval into two pair of states to calculate the likelihood. This calculation might cause additional calculation cost. Therefore, it is necessary to evaluate the calculation cost. In addition, the emission probability $b_{j,d_j}(\mathbf{o}_{1:d_j})$ can be parametric or non-parametric. In this proposal, the relation of state duration and state interval is not represented

in a model. For this reason, $b_{j,d_j}(\mathbf{o}_{1:d_j})$ is handled as non-parametric, discrete, and independent of the duration. Then, $p(L_{i,j})$ is also discrete and independent of the duration and the transition probability.

8 Evaluations

This section presents a description of the performance evaluations of models. After explaining the specification of the experimental data in Section 8.1, Section 8.2 and Section 8.3 show the experimental results of the execution time and recognition performance comparison among HSMM, IS-HSMM, and ILP-HSMM. Finally, we evaluate the reproducibility comparison between IS-HSMM and ILP-HSMM in terms of the modeling performance in Section 8.4.

8.1 Experimental Data

Addressing that the sequential data contain state duration and state interval, we use music sound data played by instruments of different kinds. When the same music is played by the different instruments, even if the music rhythm is the same, the length of each sound for the same note differs. For example, the sound power spectrum played by organ and drum for the same music sound data is shown in Figure 10. The horizontal axis represents time, and the vertical axis represents the sound power, i.e., sound volume. While the power of each note played by the organ is almost the same during the sound resonating, the one played by the drum decreases rapidly after tapping. We generate the observation sequence from the music sound data. The generation step is described below using the features of sound continuous time.

Step 1

Set the threshold b_1 and b_2 to classify the observation symbols into three types by the level of the volume. b_1 is a threshold for determining whether the sound “on” or “off”, and b_2 is the one for classifying the power of the sound into “high” and “low”. ($b_2 \geq b_1$)

Step 2

For the sound power v of each time, the observation sequence is generated as follows.

If $v \geq b_2$, then the observation symbol is “high”.

If $b_2 > v \geq b_1$, then the observation symbol is “low”.

An example of observation sequence generated by the procedure described above is shown in Figure 11. The black cell represents the “high” symbol; the gray cell represents a “low” symbol, and the cells of the white-painted represents the “interval.” To indicate the segment of a sequence, we add “start” and “end” symbols to each edge of the sequence. These symbols are useful for modeling the transition from the initial state from sequences precisely. The dataset consists of 27 segmented data, which are divided into bars of the music sequence, and each bar is assigned a label. A label is assigned for each 27 segmented data, therefore the number of labels is also 27. The kinds of the instruments are a grand piano, horn, drums, acoustic guitar, flute, and pipe organ. We use the music sound data played by the instrument of first three kinds for the training data, while the latter three kinds are used for the recognition data. The number of the sequential data is 81 for both training and recognition.

8.2 Execution Time Evaluation

This section presents the execution time evaluation for training and recognition. For the evaluation, we generate 35 sequences, fixing $d_{min} = d_{max} = 2$, $l_{min} = 1$, $l_{max} = 10$, and T is not fixed *a priori*.

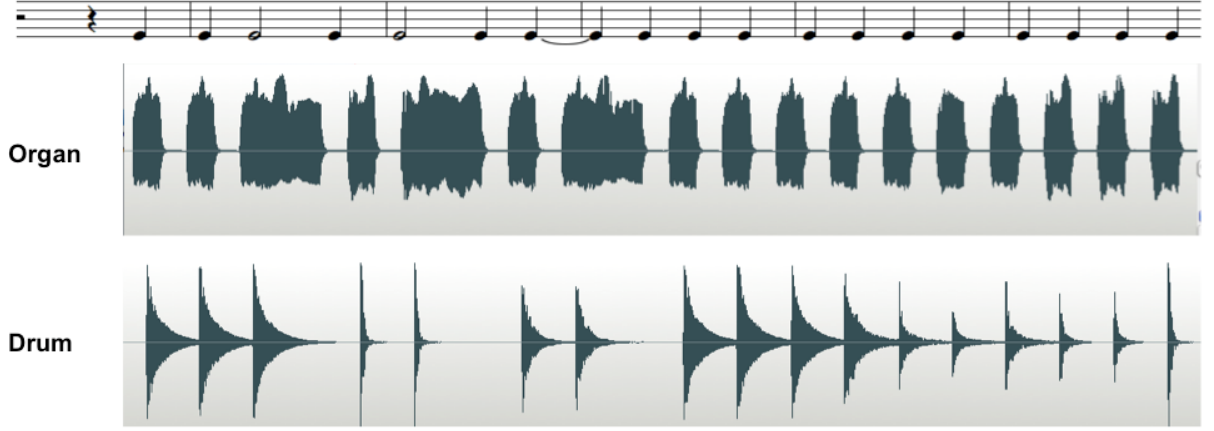


Figure 10: Music sound data.

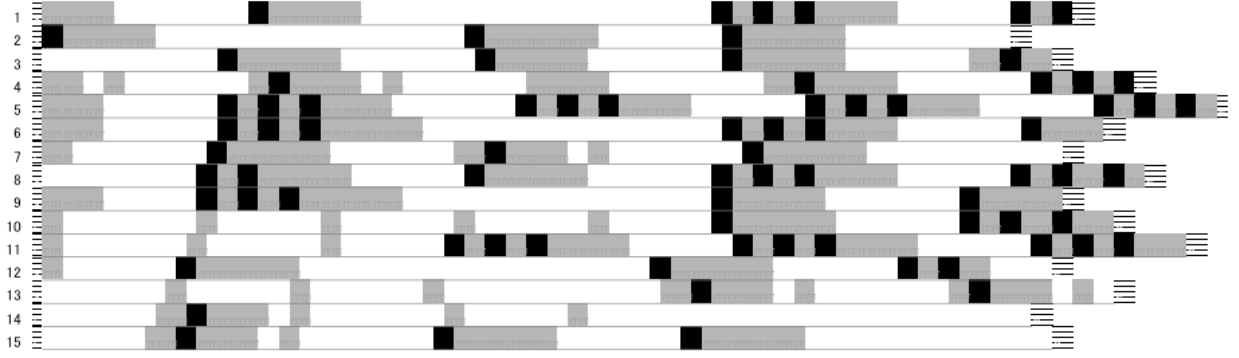


Figure 11: Example sequences generated using music sound data.

Using the generated data, we compare training time and recognition time while changing the number of training data. The training time results are shown in Figure 12. The x -axis shows the number of training data, and the y -axis shows the execution time for training. The upper, middle, and bottom lines represent the results of IS-HSMM, ILP-HSMM, and HSMM, respectively. Results show that three graphs are mostly increasing parallel, which shows that the difference between the results of HSMM and IS-HSMM, and the difference between the results of HSMM and ILP-HSMM are both of a certain degree. Therefore, the training time of IS-HSMM and ILP-HSMM requires additional time, but the amount of the additional time does not increase exponentially.

Similarly, the execution time for recognition is shown in Figure 13. The x -axis shows the number of test data, and the y -axis shows the execution time for recognition. The upper, middle, and bottom lines represent the results of IS-HSMM, ILP-HSMM, and HSMM, respectively. Results say that the amount of the additional time for recognition does not increase exponentially to the same degree as training. Stated differently, both evaluation results of training time and recognition time reveal that it does not cause any serious problems for the the execution times.

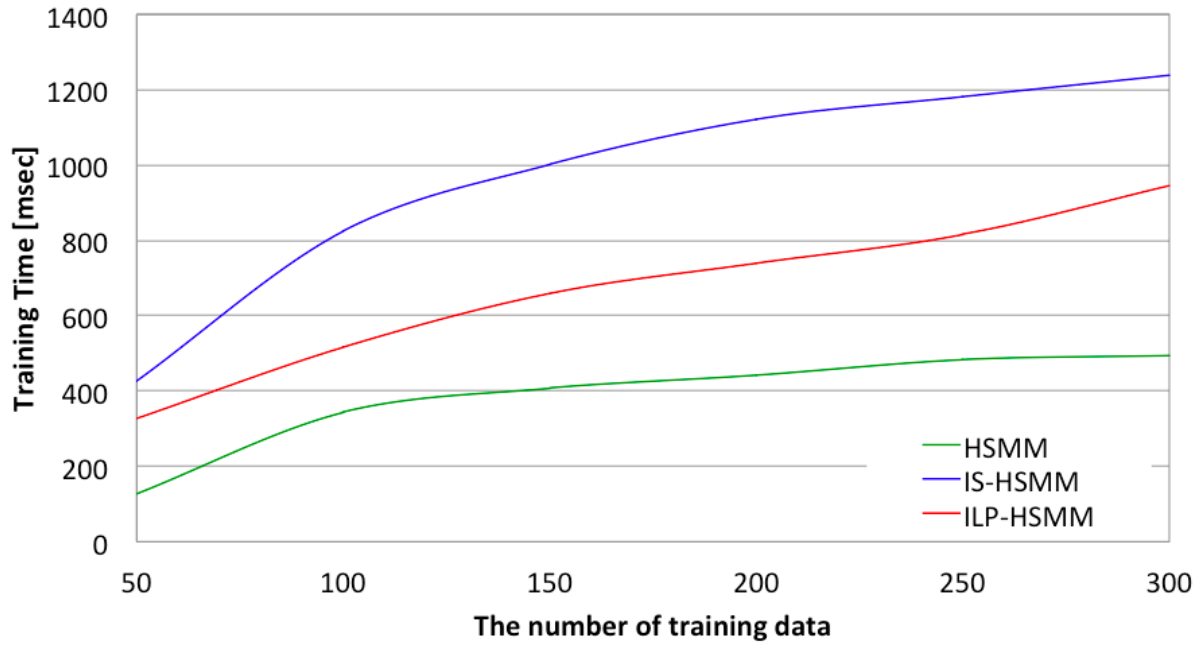


Figure 12: Execution time for training.

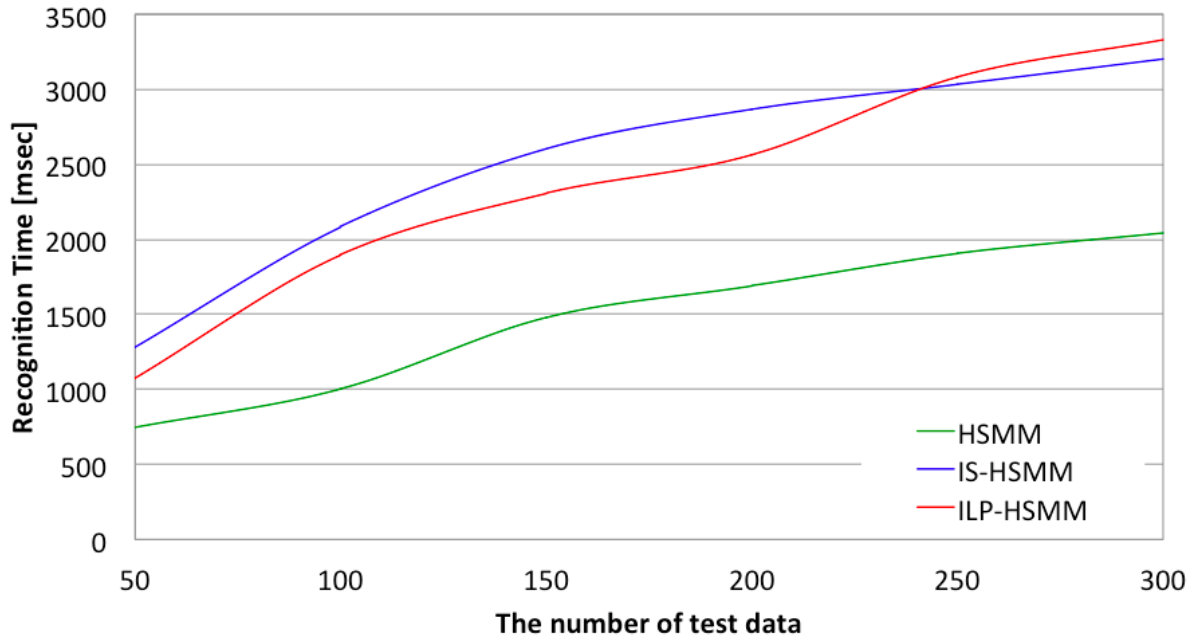


Figure 13: Execution time for recognition.

Algorithm 2 Algorithm for training and recognition in IS-HSMM.

Require: Input

Training sequences : $\mathbf{o}_{1:T_r}^z = \{o_1^z, \dots, o_{T_r}^z\}$,

Testing sequences : $\mathbf{o}_{1:T_t}^* = \{o_1^*, \dots, o_{T_t}^*\}$.

Ensure: Training phase

```
1: for  $z = 1$  to  $Z$  do
2:   Assign random values to the HSMM parameters  $\Lambda = \{A, B, \pi\}$ , and  $\alpha_{t(j,d_j)}$  and  $\beta_{t(j,d_j)}$ .
3:   for  $h = 1$  to  $H$  do
4:     for  $t = 1$  to  $T_r$  do
5:       if  $o_{t-1}$  is interval symbol then
6:         Calculate  $\alpha_{t(j,d_j)}$  and  $\beta_{t(j,d_j)}$  with joint probability from  $i$  and  $i_s$  using (7) and (8).
7:       else
8:         Calculate  $\alpha_{t(j,d)}$  and  $\beta_{t(j,d_j)}$  with preceding state  $i$  using (1) and (2).
9:       end if
10:      Update parameters  $\Lambda$ .
11:    end for
12:    Calculate  $\theta_h$  using (5) with (7).
13:    if  $\theta_h - \theta_{h-1} < \epsilon$  then
14:      break
15:    end if
16:  end for
17: end for
```

Ensure: Testing phase

```
18: for  $z = 1$  to  $Z$  do
19:   for  $t = 1$  to  $T_t$  do
20:     if  $o_{t-1}$  is interval symbol then
21:        $\Lambda^z \leftarrow$  parameter  $\Lambda$  of model  $z$  with joint probability from  $j$  and  $i_s$ .
22:     else
23:        $\Lambda^z \leftarrow$  parameter  $\Lambda$  of model  $z$  with preceding state  $j$ .
24:     end if
25:     Calculate  $\alpha_t(j, d_j)$  using (6) with (9).
26:   end for
27:   Calculate  $P(o_{1:T_t} | \Lambda^z)$  using  $\alpha_t(j, d_j)$ .
28: end for
29: Select the model  $z^*$  that has the maximum value for  $P(o_{1:T_t}^* | \Lambda^{z^*})$ .
30: Return Model  $z^*$  and its probability  $P(o_{1:T_t}^* | \Lambda^{z^*})$ .
```

Algorithm 3 Algorithm for training and recognition in ILP-HSMM.

Require: Input

Training sequences : $\mathbf{o}_{1:T_r}^z = \{o_1^z, \dots, o_{T_r}^z\}$,

Testing sequences : $\mathbf{o}_{1:T_t}^* = \{o_1^*, \dots, o_{T_t}^*\}$.

Ensure: Training phase

```
1: for  $z = 1$  to  $Z$  do
2:   Assign random values to the HSMM parameters  $\Lambda = \{A, B, \pi, L\}$ , and  $\alpha_{t(j,d_j)}$  and  $\beta_{t(j,d_j)}$ .
   Initialize  $p(L_{i,j})$  as  $L_{i,j} = 1$ .
3:   for  $h = 1$  to  $H$  do
4:     for  $t = 1$  to  $T_r$  do
5:       Calculate  $\alpha_{t(j,d_j)}$  and  $\beta_{t(j,d_j)}$  using (1) and (2).
6:       Calculate  $p(L_{i,j})$  with  $i$  and  $j$  using (10).
7:       Update parameters  $\Lambda$ .
8:     end for
9:     Calculate  $\theta_h$  using (9).
10:    if  $\theta_h - \theta_{h-1} < \epsilon$  then
11:      break
12:    end if
13:  end for
14: end for
```

Ensure: Testing phase

```
15: for  $z = 1$  to  $Z$  do
16:   for  $t = 1$  to  $T_t$  do
17:      $\Lambda^z \leftarrow$  parameter  $\Lambda$  of model  $z$ .
18:      $p(l) \leftarrow p(L_{i,j})$  using  $\Lambda^z$  with observed interval  $l$ .
19:     Calculate  $\alpha_t(j, d_j)$  using (6) with (11).
20:   end for
21:   Calculate  $P(o_{1:T_t} | \Lambda^z)$  using  $\alpha_t(j, d_j)$ .
22: end for
23: Select the model  $z^*$  that has the maximum value for  $P(\mathbf{o}_{1:T_t}^* | \Lambda^z)$ .
24: Return Model  $z^*$  and its probability  $P(\mathbf{o}_{1:T_t}^* | \Lambda^{z^*})$ .
```

8.3 Recognition Performance Evaluation

This section presents the evaluation results of recognition performance comparing IS-HSMM and ILP-HSMM with HSMM. The evaluation metric is the recognition accuracy based on f -measure that is calculated using $(2 \cdot \text{recall} \cdot \text{precision}) / (\text{recall} + \text{precision})$, where $\text{precision} = TP/PP$, and $\text{recall} = TP/AP$. Here, when the Predicted Positive (PP) is the number of models with likelihood calculated using (6) is maximum in all models, True Positive (TP) is the number of collected models in PP , and Actually Positive (AP) is the number of labeled models.

Results are presented in Figure 14 and Figure 15. The x -axis represents Precision, Recall, and f -measure, and the y -axis represents the score. The left, middle and right bars represent the results of HSMM, IS-HSMM, and ILP-HSMM, respectively. Figure 14 shows the results obtained when the number of states is 5, and Figure 15 presents the results obtained when the number of states is 10. Both results are the average scores of five repetitions. The results showed that both the proposed models IS-HSMM and ILP-HSMM have higher recognition performance than HSMM. By comparing the results of IS-HSMM and ILP-HSMM, the scores of f -measure are similar, but the scores of recall and precision differ. IS-HSMM has a higher score for recall, but it has lower score for precision than ILP-HSMM. In the next section, we analyze the performance of IS-HSMM and ILP-HSMM in detail. Finally, comparison of the two results obtained when the numbers of states are 5 and 10 shows that the recognition performance can be higher depending on the number of states increasing.

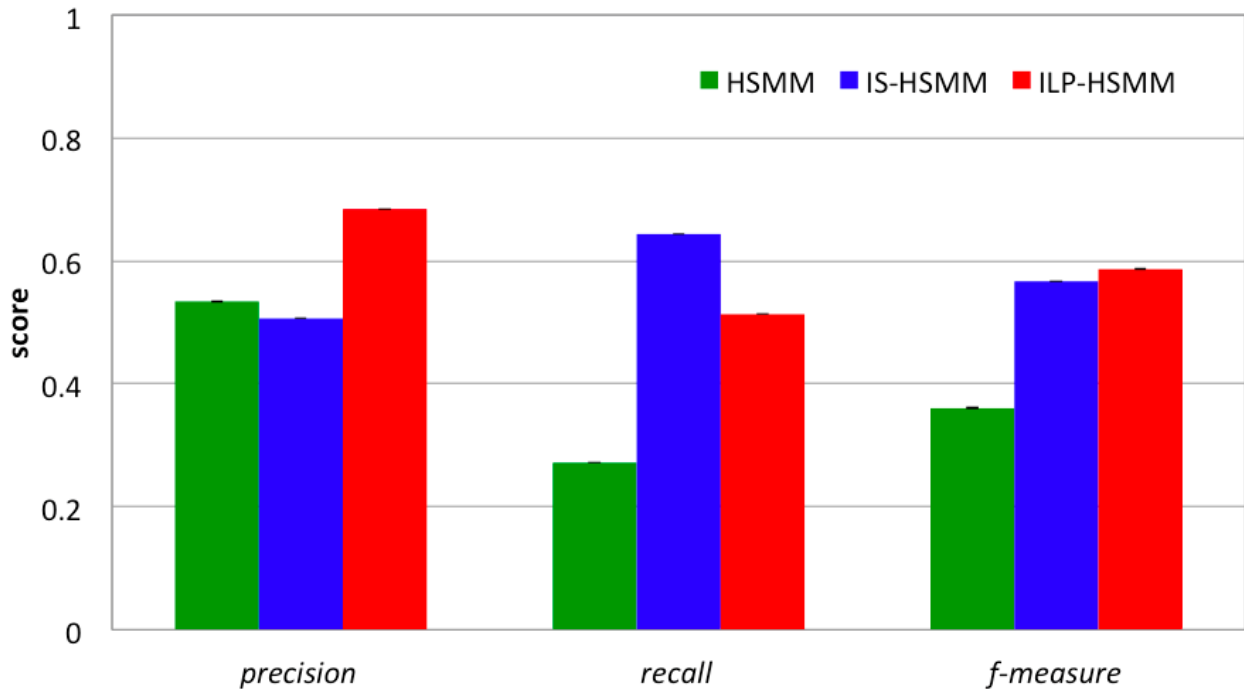


Figure 14: Recognition performance: the number of states is 5.

The earlier experiment includes observation symbols of only three kinds. To evaluate the performance of treating various durations and intervals with observation symbols of many kinds, we use the musical scale instead of the volume of the sound as observation symbols. Figure 16 shows the musical scale with stairs of example data. These are the some input data extracted from the

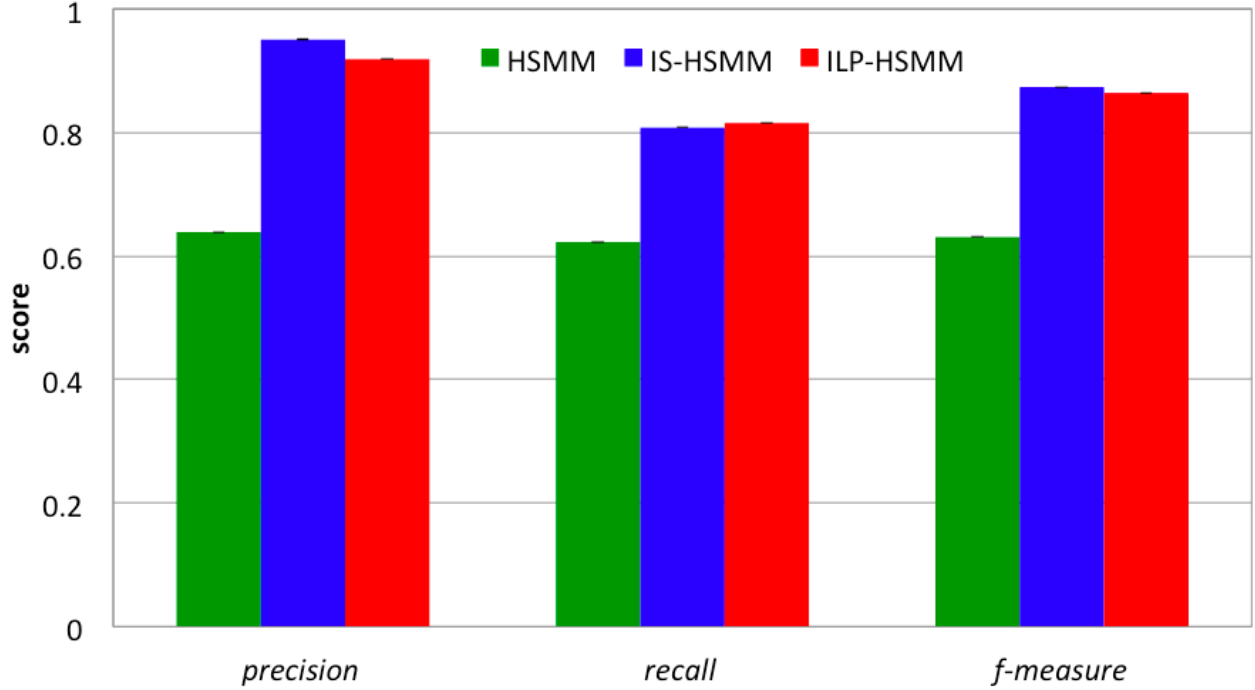


Figure 15: Recognition performance: the number of states is 10.

evaluation data. The number on the each graph signifies the label. Each value from 0.01 to 0.12 in 0.01 intervals is assigned to C, C#, D, D#, E, F, F#, G, G#, A, A#, B of the musical scale. If the volume is lower than a threshold, then the value of sound scale label is zero. It means the *interval observation* in a sequence. The results of recognition performance using the data generated as described above are shown in Figure 17 and Figure 18. They show results of recognition performance evaluation when the numbers of states are 2 and 10. The scores are the average scores of five repetitions. Considering that it would be high performance when the number of states is greater than the number of observation symbols in HSMM, we assign 2 and 10 as the number of states in the experience to compare the performance of them.

When the number of states is 2, the recognition performance of HSMM is extremely low, but those of IS-HSMM and ILP-HSMM are much higher than HSMM. In addition to this, the results of IS-HSMM are much higher than ILP-HSMM. However, when the number of states is 10, the number of states is greater than the number of the observation symbols. At this time, the entire scores of HSMM, IS-HSMM, and ILP-HSMM are higher than 0.4. For the HSMM, the recall score gives the max score in all models but the precision score indicates the lowest value. Therefore, the probability for each sequence using HSMM is similar to that of each other sequence. Then, while the average scores of precision, recall, and f -measure are more than 0.8 in IS-HSMM, the average score is about 0.7 in ILP-HSMM. As a result, when the number of states increases, the scores of IS-HSMM are higher than those of ILP-HSMM because increasing the states contributes to treatment of the transition probability from a state to another state. Therefore, IS-HSMM is effective for treating the order of the sequence precisely because the “interval” is represented with one of states and HMM can model the transition probability between two states.

However, regarding the input data shown in Figure 16 in detail, No. 4 input data are similar to No. 7; the No. 2 input data are similar to No. 10. It is difficult to distinguish the small time

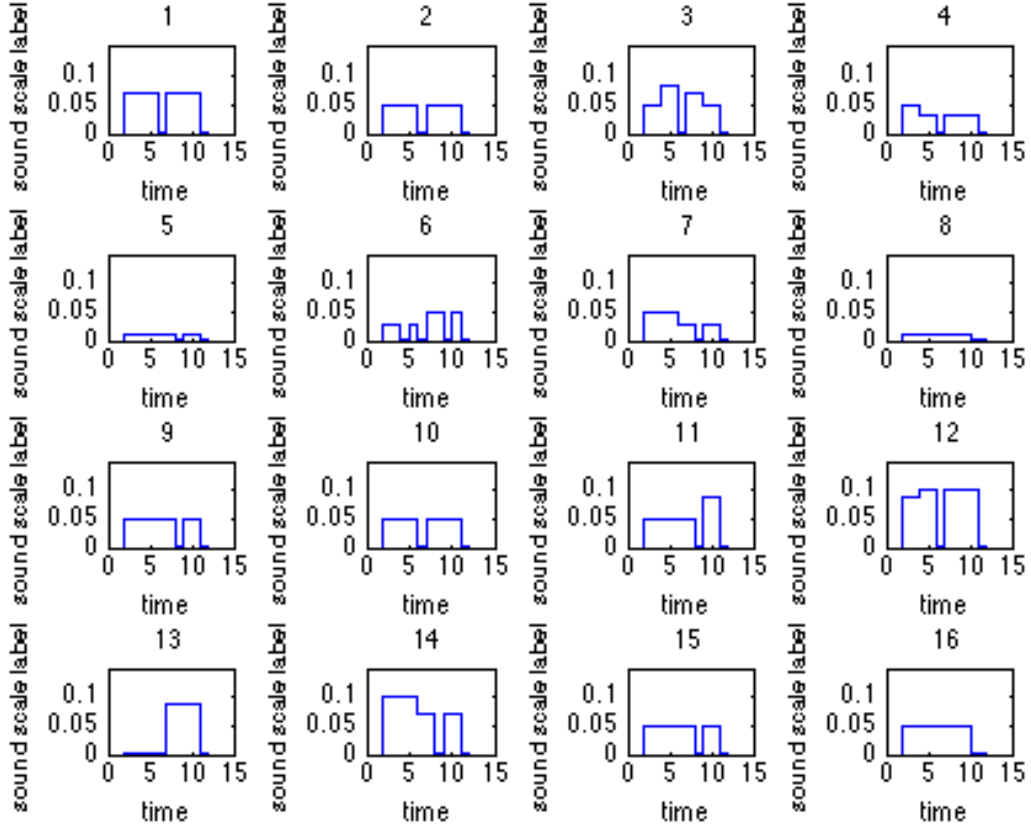


Figure 16: Musical scale of example input sequences and its label.

difference between two sequences with both IS-HSMM and ILP-HSMM even if the number of states increases. It might cause the decline of recognition performance.

Moreover, ILP-HSMM treats the state interval using the new additional parameter between two stationary states. If state interval is mostly similar between static two states, ILP-HSMM can model the length of interval precisely, but it is difficult to model a sequence including various length of duration and interval. Therefore, to treat the sequential data of various kinds with duration and interval, IS-HSMM would lead to higher performance than ILP-HSMM. The following section shows the evaluation results of modeling performance and analysis between ILP-HSMM and IS-HSMM.

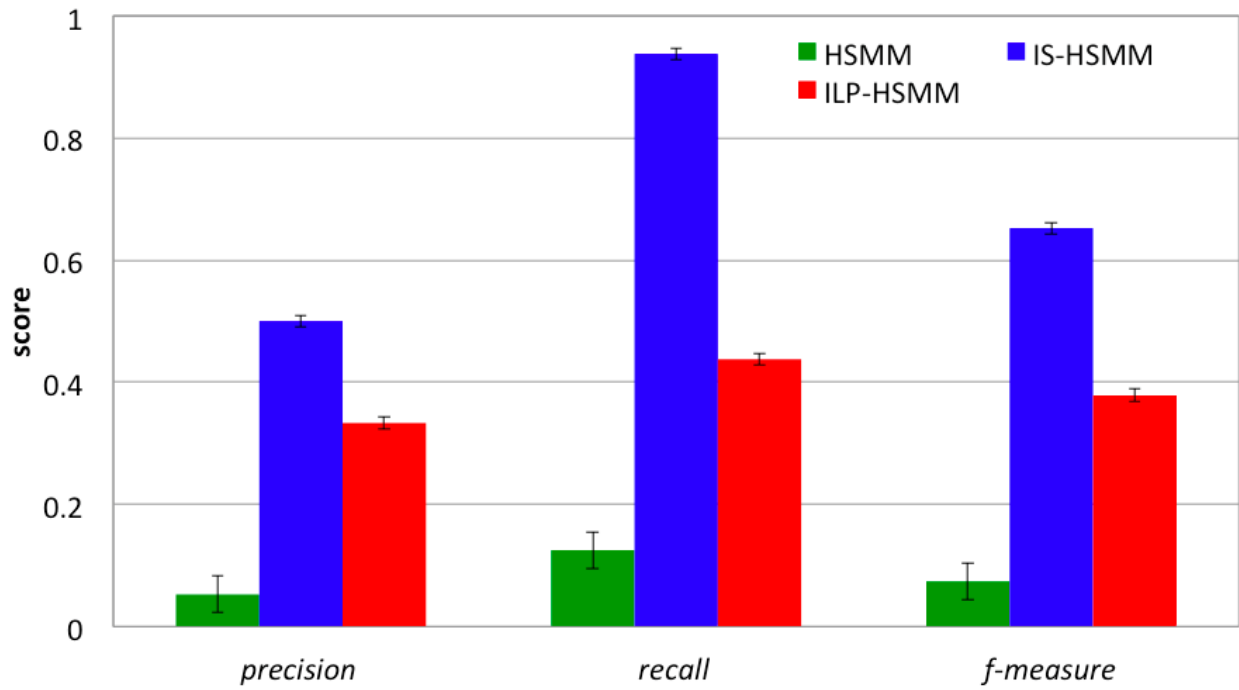


Figure 17: Recognition performance with music scale label: the number of states is 2.

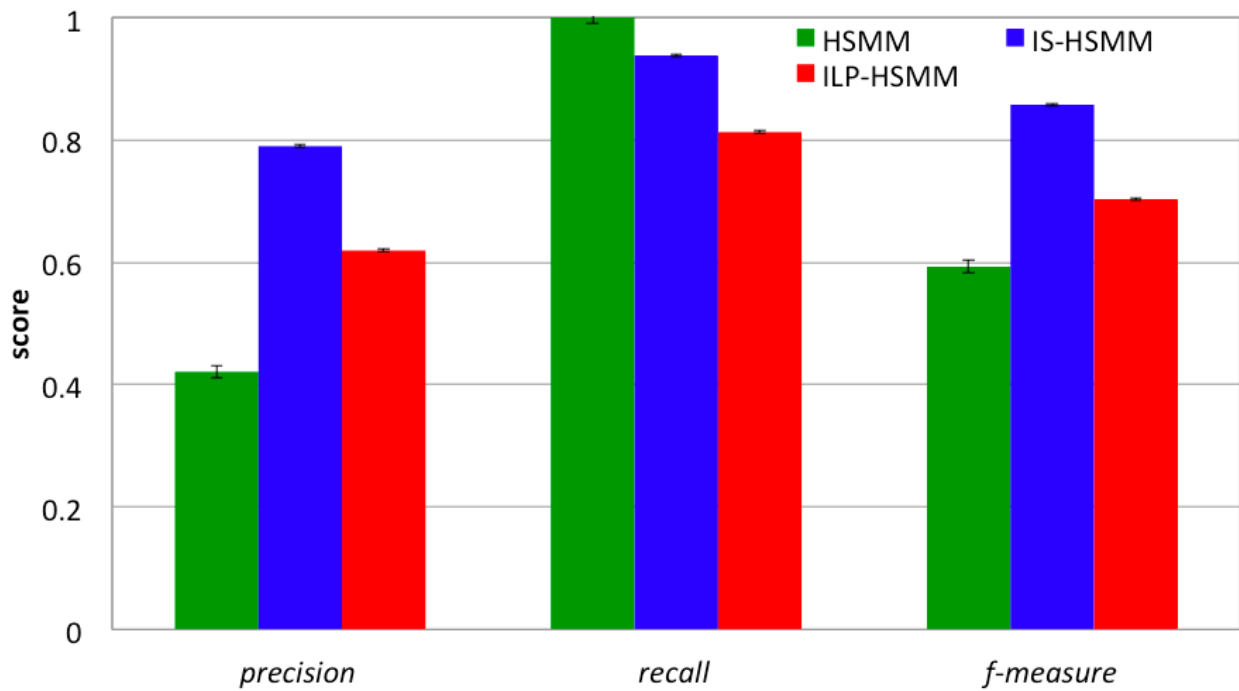


Figure 18: Recognition performance with music scale label: the number of states is 10.

8.4 Reproducibility Performance Evaluations between IS-HSMM and ILP-HSMM

This section presents the evaluation results of modeling performance, particularly addressing the performance of reproducibility. We calculate the performance of reproducibility and compare both IS-HSMM and ILP-HSMM. The percentage modeling performance indicates how precisely the model generates the original sequence, which is represented as r . The modeling performance r is calculated as

$$r = \frac{\sum_{t=1}^T (w_t = o_t)}{T},$$

where $\mathbf{o}_{1:T}$ is the original sequence, T is the time length of the original sequence, and $w_{1:T}$ is the generated sequence using the model parameter θ which is calculated using the original sequence. To calculate the equation presented above, we give the sequence length T and generate a sequence which has high likelihood using the forward algorithm with the set of parameters Λ . The generated sequence is the estimated sequence. Therefore, the performance of reproducibility indicates how precisely the model, i.e., the set of parameters Λ decided by the training phase, generates the original sequence.

First, we evaluate the performance of reproducibility when the number of states changes. Figure 19 presents the results of evaluating reproducibility using the HSMM, IS-HSMM and ILP-HSMM. The x -axis represents the number of states, and the y -axis represents the performance of reproducibility. The number of observed symbols in a sequence N is $N = 7$.

Results show that all of the models obtain higher performance of reproducibility when the number of states increases. The performance results of IS-HSMM and HSMM is mostly the same and IS-HSMM has a bit higher performance than that of HSMM. The results of ILP-HSMM show less performance when the number of states is under six, and those show higher performance when the number of states is over six. It represents that the number of states is more than the number of observed symbols, ILP-HSMM has higher performance of reproducibility than other models.

Then, we evaluate the performance of reproducibility when the number of intervals in a sequence changes. Figure 20 also shows the scores of performance of reproducibility of HSMM, IS-HSMM and ILP-HSMM. The x -axis represents the number of intervals in a sequence, and the y -axis represents the score of modeling performance. The number of sorts observed in a sequence is $N = 6$. One of the sorts is an interval. Results show that the performance of reproducibility of both models; HSMM and IS-HSMM decrease as the number of intervals increases, but that of IS-HSMM is higher than that of HSMM. Then, the results of ILP-HSMM is the highest performance in all models and it can obtain the highest performance regardless of the number of intervals. Therefore, IS-HSMM can model the sequence with intervals more precisely than HSMM, and ILP-HSMM can model it most precisely in all models. Comparing two results; HSMM and IS-HSMM, it ensures that the proposed IS-HSMM can model the sequential data more precisely than HSMM by introducing the special state, i.e., the interval state and calculating the transition probability from the state prior to the interval state. In addition, the performance of IS-HSMM is much higher especially when the states are few and even if many intervals exist in a sequence. Comparing the other results; IS-HSMM and ILP-HSMM, it ensures that the proposed ILP-HSMM can model the sequential data more precisely than other models because it can represent the length of intervals directly in the model.

As a result of evaluation above, both the proposed extension models for HSMM have higher performance than HSMM, but ILP-HSMM can model the static interval between two states. However, it is more important for modeling the general duration and interval using a model trained

multiple data which have the same label. From the view point of modeling generalization, the recognition performance of IS-HSMM has a higher score than other models, especially where the number of sorts of the observation symbols is larger. Therefore, we conclude that IS-HSMM has higher performance for modeling the general sequential data, not only for the data which have a static length of interval, but also for data which have various interval lengths.

9 Summary and Future Work

The goal of this research was to model sequential data, including state duration and state interval, simultaneously. We specifically examined a hidden semi-Markov model (HSMM) to treat such sequential data, and propose two extended models to treat state interval in a sequence: IS-HSMM and ILP-HSMM. IS-HSMM introduces a special calculation technique to treat interval state, where if the preceding state is an interval state, it models the transition from the second preceding state to the current state simultaneously. On the other hand, ILP-HSMM uses the Gaussian distribution as a length parameter, and trains with both preceding and subsequent states. From the comparisons of recognition performance and elapsed time among IS-HSMM, ILP-HSMM, and HSMM, both of the proposed models give higher performance than HSMM although they need additional calculation costs. The comparison results between IS-HSMM and ILP-HSMM in terms of the modeling performance reveal that ILP-HSMM has higher performance than that of IS-HSMM.

As the future directions, we intend to adopt our model to treat such actual sensing data which have a feature of rhythm or timing patterns. While ILP-HSMM has higher performance in the evaluation, the concept of IS-HSMM is simpler than that of ILP-HSMM. Additionally, IS-HSMM can adopt another problem of analyzing the sequential data except for only treating intervals between states. In case that the same state occurs frequently in a sequence, it is difficult to model the original sequence without interval precisely. Therefore, we need to evaluate the effectiveness of treating the original sequence using other application data, and finally extend the model further.

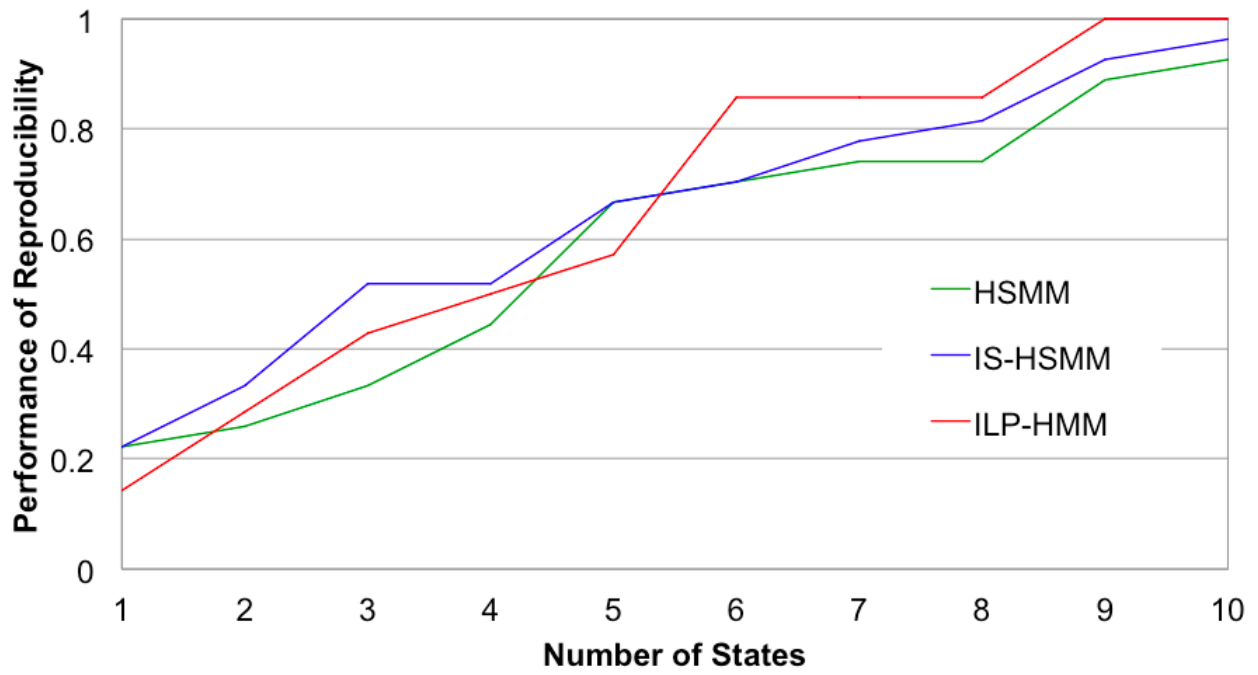


Figure 19: Modeling performance when the number of intervals increases.

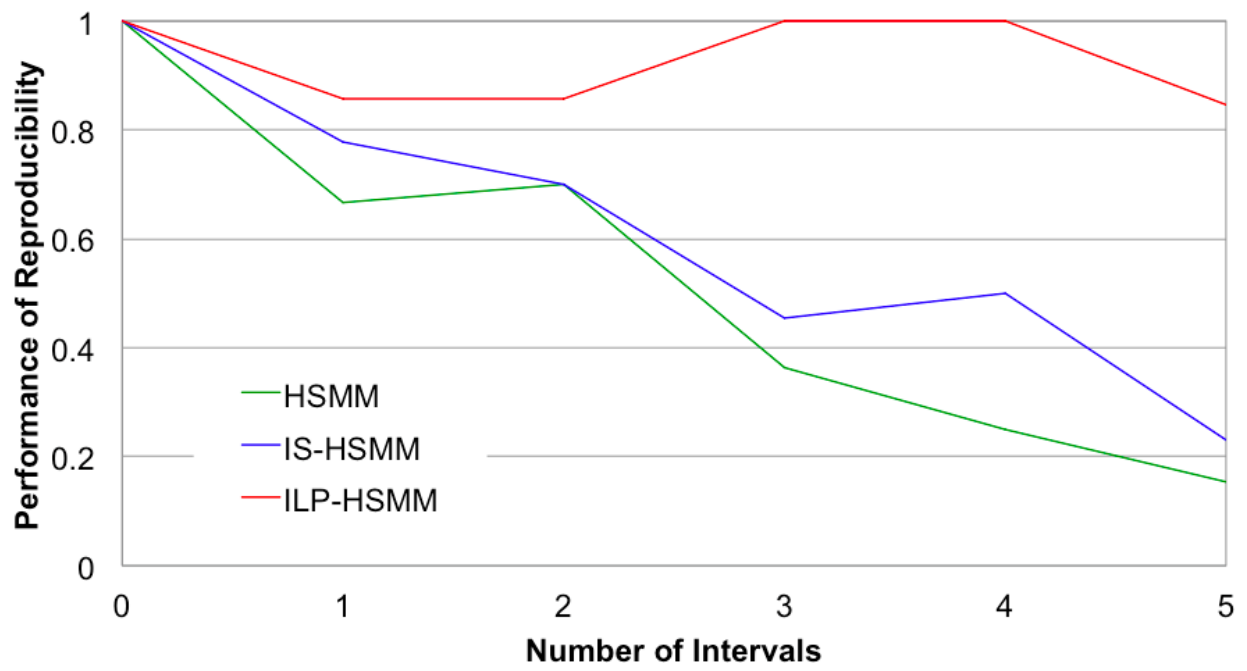


Figure 20: Modeling performance when the intervals become more numerous.

References

- [1] M. Esmaeili and F. Gabor, “Finding sequential patterns from large sequence data,” *International Journal of Computer Science Issues (IJSC)*, vol. 7, no. 1, pp. 43–46, 2010.
- [2] D. D. Lewis and W. A. Gale, “A sequential algorithm for training text classifiers,” *Proc. of ACM the 17th Annual International Conference on Research and Development in Information Retrieval (ACM SIGIR)*, pp. 3–12, 1994.
- [3] J. Song and D. L. Nicolae, “A sequential clustering algorithm with application to gene expression data,” *Journal of the Korean Statistical Society*, vol. 38, no. 2, pp. 175–184, 2009.
- [4] H. Banaee, M. U. Ahmed, and A. Loutfi, “Data mining for wearable sensors in health monitoring systems: A review of recent trends and challenges,” *Sensors 2013*, vol. 13, no. 12, pp. 17 472–17 500, 2013.
- [5] Y. Zheng, R. Niu, and P. K. Varshney, “Sequential bayesian estimation with censored data for multi-sensor systems,” *IEEE Trans. on Signal Processing*, vol. 62, no. 10, pp. 2626–2641, 2014.
- [6] H. T. Cheng, “Learning and recognizing the hierarchical and sequential structure of human activities,” Ph.D. dissertation, Carnegie Mellon University, Dec. 2013.
- [7] S. Z. Yu, “Hidden semi-markov models,” *Elsevier Artificial Intelligence*, vol. 174, no. 2, pp. 215–243, 2010.
- [8] H. Narimatsu and H. Kasai, “Duration and interval hidden markov model for sequential data analysis,” *Proc. of International Joint Conference on Neural Networks (IJCNN2015)*, pp. 3743–2751, 2015.
- [9] H. Sakoe and S. Chiba, “A dynamic programming approach to continuous speech recognition,” *Proc. of 7th International Congress on Acoustics (ICA) 1971*, vol. C13, 1971.
- [10] V. N. Vapnik, “Statistical learning theory,” *John Wiley and Sons, New York*, 1995.
- [11] S. Abe, “Support vector machines for pattern classification,” *Springer Science and Business Media*, July 2010.
- [12] W. J. Boscardin and A. Gelman, “Bayesian regression with parametric models for heteroscedasticity,” *Advances in Econometrics*, vol. 11A, pp. 87–109, 1996.
- [13] D. R. Cox, “The regression analysis of binary sequences,” *Journal of the Royal Statistical Society*, vol. 20, pp. 215–242, 1958.
- [14] L. E. Baum and T. Petrie, “Statistical inference for probabilistic functions of finite state markov chains,” *The Annals of Mathematical Statistics*, vol. 37, no. 6, pp. 1554–1563, 1966.
- [15] L. E. Baum and J. A. Egon, “An inequality with applications to statistical estimation for probabilistic functions of a markov process and to a model for ecology,” *Bulletin of the American Mathematical Society*, vol. 73, no. 3, pp. 360–363, 1967.
- [16] H. Xue and V. Govindaraju, “Hidden markov models combining discrete symbols and continuous attributes in handwriting recognition,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 28, no. 3, pp. 458–462, 2006.

- [17] Y. Bengio and P. Frasconi, "Input-output hmm for sequence processing," *IEEE Trans. on Neural Networks*, vol. 7, no. 5, 1996.
- [18] F. Salzenstein, C. Collet, S. Lecam, and M. Hatt, "Non-stationary fuzzy markov chain," *Pattern Recognition Letters*, vol. 28, no. 16, pp. 2201–2208, 2007.
- [19] S. Z. Yu and H. Kobayashi, "An efficient forward-backward algorithm for an explicit duration hidden markov model," *IEEE Signal Processing Letters*, vol. 10, no. 1, pp. 11–14, 2003.
- [20] C. D. Mitchell and L. H. Jamieson, "Modeling duration in a hidden markov model with the exponential family," *Proc. of 1993 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 2, pp. 331–334, Apr. 1993.
- [21] P. Ramesh and J. G. Wilpon, "Modeling state duration in hidden markov models for automatic speech recognition," *Proc. of 1992 IEEE International Conference on Acoustic, Speech, and Signal Processing (ICASSP)*, vol. 1, pp. 381–384, Mar. 1992.
- [22] K. Murphy, "Dynamic bayesian networks: Representation, inference and learning," Ph.D. dissertation, Dept. Computer Science, UC Berkeley, 2002.
- [23] —, "Hidden semi-markov models (hsmms)," <http://www.ai.mit.edu/murphyk>, Nov. 2002, (accessed 2016-02-19).
- [24] J. Bulla, "Application of hidden markov models and hidden semi-markov models to financial time series," Ph.D. dissertation, Georg-August-University of Gottingen, June 2006.
- [25] M. Dong and D. He, "Hidden semi-markov model-based methodology for multi-sensor equipment health diagnosis and prognosis," *European Journal of Operational Research*, vol. 178, no. 3, pp. 858–878, April 2006.
- [26] N. A. Dasu, "Implementation of hidden semi-markov models," Ph.D. dissertation, University of Nevada, May 2011.
- [27] S. R. Eddy, "Multiple alignment using hidden markov models," *Proc. of AAAI Third International Conference on Intelligent Systems for Molecular Biology*, vol. 3, pp. 114–120, 1995.
- [28] Y. B. Yoma and J. S. Sanchez, "Map speaker adaptation of state duration distribution for speech recognition," *IEEE Trans. on Speech and Audio Processing*, vol. 10, no. 7, pp. 443–450, 2002.
- [29] Y. K. Park, C. K. UN, and O. W. Kwon, "Modeling acoustic transitions in speech by modified hidden markov models with state duration and state duration-dependent observation probabilities," *IEEE Trans. on Speech and Audio Processing*, vol. 4, no. 5, pp. 389–392, 1996.
- [30] H. Kobayashi and S. Z. Yu, "Hidden semi-markov models and efficient forward-backward algorithms," *Proc. of 2007 Hawaii and SITA Joint Conference on Information Theory*, vol. 174, pp. 41–46, May 2007.